CSCI 253

Object Oriented Design:
Java Review – Errors
George Blankenship

Java Review Topics

- Errors ←
- Exceptions
- Debugging

Program Errors

- Types of errors
  - Compile-time (syntax) errors
  - Run-time errors
  - Logic errors

Program Errors – Compile Time

- Compile-time (syntax) errors
  - Errors in code construction
    - Lexical (typographical), grammatical, types
  - Detected during compilation
  - Usually easy to correct quickly
- Examples
  - Misspelled keyword
  - Missing or misplaced symbol
  - Incorrect operator for variable type

Java Review - Errors George Blankenship 4

Program Errors – Run Time

- Run-time errors
  - Operations illegal / impossible to execute
  - Detected during program execution
  - Treated as exceptions in Java
- Example
  - Division by zero
  - Array index out of bounds
  - Using null pointer
  - Illegal format conversion

Java Review - Errors George Blankenship 5

Program Errors – Logic

- Logic errors
  - Operations leading to incorrect program state
  - May (or may not) lead to run-time errors
  - Problem in design or implementation of algorithm
- Examples
  - Computing incorrect arithmetic value
  - Ignoring illegal input
- Hardest error to handle
- Detect by testing, debugging

Java Review - Errors George Blankenship 6

Java Review Topics

- Errors
- Exceptions ←
- Debugging

Exception

- Rare event outside normal behavior of code - anomaly
- Examples
  - Division by zero
  - Access past end of array
  - Out of memory
  - Number input in wrong format (float vs. integer)
  - Unable to write output to file
  - Missing input file

Exception Handling

- Perform action in response to exception
  - Ignore exception
  - Print error message
  - Request new data
  - Retry action
- Approaches
  - Exit program
  - Exit method returning error code
  - Throw exception

Exit Program

- Exit program with error message / error code
- Example

```
if (error) {
    System.err.println("Error found");     // message
    System.exit(1);                        // error code
}
```

- Problem
  - Drastic solution
  - Event must be handled by user invoking program
  - Program may be able to deal with some exceptions

Error Code

- Exit function with return value ⇒ error code
- Example

```
A( ) { if (error) return (-1); }
B( ) { if ((retval = A( )) == -1) return (-1); }
```

- Problems
  - Calling function must check & process error code
    - May forget to handle error code
    - May need to return error code to caller
  - Agreement needed on meaning of error code
  - Error handling code mixed with normal code

Throw Exception

- Approach
  - Throw exception to signal anomaly
- Example

```
A( ) {
    if (error) throw new ExceptionType(…);
}
B( ) {
    try {
      A( );
      }
    catch (ExceptionType e) { ...action... }
}
```

Java exception backtracks to caller(s) until matching catch block found

Representing Exceptions in Java

- Exceptions represented as
  - Objects derived from class Throwable
- Code

```
public class Throwable( ) extends Object {
   Throwable( )                      // No error message
   Throwable( String mesg )          // Error message
   String getMessage()               // Return error mesg
   void printStackTrace( ) { … }     // Record methods called & location
   …
}
```

Java Review - Errors George Blankenship 13

Generating and Handling Exceptions

- Java primitives
  - Try
  - Throw
  - Catch
  - Finally
- Procedure for using exceptions
  - Enclose code generating exceptions in try block
  - Use throw to actually generate exception
  - Use catch to specify exception handlers
  - Use finally to specify actions after exception

Java Review - Errors George Blankenship 14

Java Syntax

```
try {                          // try block encloses throws
   throw new eType1();         // throw jumps to catch
}
catch (eType1 e) {             // catch block 1
   ...action...                // run if type match
}
catch (eType2 e) {             // catch block 2
   ...action...                // run if type match
}
finally {                      // final block
   ...action...                // always executes
}
```

Java Review - Errors George Blankenship 15

Exceptions – Examples

- FileNotFoundException (java.io)
  - Request to open file fails
- IllegalArgumentException (java.lang)
  - Method passed illegal / inappropriate argument
- IOException (java.io)
  - Generic I/O error
- NullPointerException (java.lang)
  - Attend to access object using null reference
- UnsupportedOperationException (java.lang)
  - Object does not provide requested operation

Java Review - Errors George Blankenship 16

Designing & Using Exceptions

- Use exceptions only for rare events
  - Not for common cases ⇒ checking end of loop
  - High overhead to perform catch
- Place statements that jointly accomplish task into single try/catch block
- Use existing Java Exceptions if possible
- Avoid simply catching and ignoring exceptions
  - Poor software development style
  - Ignores those exceptions that indicate a real problem

Java Review - Errors George Blankenship 17

Java Review Topics

- Errors
- Exceptions
- Debugging ←

Java Review - Errors George Blankenship 18

Debugging

- Process of finding and fixing software errors
- Goal
  - Determine cause of run-time & logic errors
  - Correct errors (without introducing new errors)
- Similar to detective work
  - Carefully inspect information in program
    - Code
    - Values of variables
    - Program behavior

Debugging – Approaches

- Static debugging
  - Insert debugging statements
  - Trace program control flow
  - Display value of variables
- Interactive debugging
  - IDE (integrated development environment)
  - Interactive debugger

Static Debugging

- Trace statements
  - Program trace written to file
  - Program trace or actions displayed on GUI
- Problem
  - Inserted at code development
  - Code coverage may be inappropriate
  - Program execution may be impacted
- Strength
  - Can be used during normal program execution
  - Can be designed to handle all program types

Interactive Debugging

- Provides trace of program execution
- Shows location in code where error encountered
- Interactive program execution
  - Single step through code
  - Run to breakpoints
- Displays values of variables
  - For current state of program
- Problems
  - Cannot be used for normal program execution
  - Cannot cover all program types (multithread)
- Strength
  - Can be used without special code

Java Review - Errors George Blankenship 22